Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (previously presented) A method for performing operations within a file system in which directories and files are organized as nodes in a namespace tree, the method comprising:

associating a read-write lock with each of the nodes in the namespace tree;

acquiring a first lock on a name of one or more directories involved in a first operation;

acquiring a second lock on an entire pathname involved in the first operation;

determining whether the first lock or the second lock conflicts with third locks acquired by a second operation;

performing the first operation when the first lock or the second lock does not conflict with the third locks, where the first, second, and third locks are read-write locks; and

serializing performance of the first and second operations when the first lock or the second lock conflicts with the third locks.

2.   (original) The method of claim 1, wherein the performing the first operation includes:

concurrently performing the first operation and the second operation when neither the first lock nor the second lock conflicts with the third locks.

3.   (original) The method of claim 1, wherein the first lock is a read lock.

4.   (previously presented) The method of claim 1, wherein the second lock is one of a read lock or a write lock.

5.   (original) The method of claim 1, wherein the first operation is a read operation, the first lock is a read lock, and the second lock is a read lock.

6.   (original) The method of claim 1, wherein the first operation is a namespace modification operation, the first lock is a read lock, and the second lock is a write lock.

7.   (original) The method of claim 1, wherein the first operation is a snapshot operation, the first lock is a read lock, and the second lock is a write lock.

8.   (original) The method of claim 1, wherein the determining whether the first lock or the second lock conflicts with third locks includes:

        using a lazily allocated data structure that maps pathnames to locks to determine whether the first lock or the second lock conflicts with the third locks.

9.   (previously presented) The method of claim 1, wherein serializing performance of the first and second operations includes:

        serializing the first, second, and third locks when the first lock or the second lock conflicts with the third locks.

10. (currently amended) The method of claim [[9]] 1, wherein the serializing ~~the first, second, and third locks~~ performance of the first and second operations includes:

determining an order for the first, second, and third locks based on levels of the namespace tree involved in the first, second, and third locks and within one of the levels of the namespace tree involved in the first, second, and third locks.

11. (currently amended) The method of claim [[9]] 1, wherein the serializing ~~the first, second, and third locks~~ performance of the first and second operations includes:

determining an order for the first, second, and third locks, and

permitting the first and second operations to be performed based on the determined order for the first, second, and third locks.

12. (original) The method of claim 1, further comprising:

permitting the first and second operations to concurrently operate within a same one of the directories when neither the first lock nor the second lock conflicts with the third locks.

13. (previously presented) A system for performing operations within a file system, comprising:

means for obtaining one or more first locks on one or more directory names involved in an operation;

means for obtaining a second lock on an entire pathname involved in the operation;

means for detecting whether the one or more first locks or the second lock conflict with one or more third locks acquired by another operation;

means for executing the operation when the one or more first locks or the second lock do not conflict with the one or more third locks, the one or more first locks, the second lock, and the one or more third locks being read-write locks; and

means for serializing execution of the operation and the other operation when at least one of the one or more first locks or the second lock conflicts with the one or more third locks.

14.   (previously presented) A file system, comprising:

a memory configured to store information regarding directories and files organized as nodes in a namespace tree; and

a processor connected to the memory and configured to:

associate a read-write lock with each of the nodes in the namespace tree,

acquire one or more first locks on names of one or more of the directories involved in a first operation,

acquire a second lock on an entire pathname involved in the first operation,

determine whether the one or more first locks or the second lock conflict with one or more third locks acquired by a second operation,

permit the first operation to execute when the one or more first locks or the second lock do not conflict with the one or more third locks, the one or more first locks, the second lock, and the one or more third locks being read-write locks, and

serialize execution of the first and second operations when at least one of the one

5

or more first locks or the second lock conflicts with the one or more third locks.

15.   (previously presented) A method for performing first and second operations within a file system, comprising:

acquiring one or more first locks on one or more first directory names involved in the first operation;

acquiring one or more second locks on one or more second directory names involved in the second operation;

acquiring a third lock on a first pathname involved in the first operation;

acquiring a fourth lock on a second pathname involved in the second operation;

determining whether the first and third locks conflict with the second and fourth locks;

concurrently performing the first and second operations when the first and third locks do not conflict with the second and fourth locks, the one or more first locks, the one or more second locks, the third lock, and the fourth lock being read-write locks; and

serializing performance of the first and second operations when the first lock or the third lock conflicts with the second lock or the fourth lock.

16.   (previously presented) A method for performing first and second operations within a same directory, comprising:

obtaining a first lock on a sub-directory or file name within the directory by the first operation;

obtaining a second lock on a sub-directory or file name within the directory by the

second operation;

determining whether the first and second locks conflict;

concurrently performing the first and second operations when the first and second locks do not conflict, the first and second locks being read-write locks; and

serializing performance of the first and second operations when the first and second locks conflict.

17.	(previously presented) A file system, comprising:

a memory configured to store information regarding a plurality of directories and files as nodes in a namespace tree; and

a processor connected to the memory and configured to:

associate a read-write lock with each of the nodes in the namespace tree,

identify a set of the nodes involved in an operation, the identified nodes forming a pathname associated with the operation,

acquire a first one or more read-write locks, as one or more first locks, on the identified nodes,

acquire a second one of the read-write locks, as a second lock, on the pathname,

determine whether the one or more first locks or the second lock conflict with a read-write lock acquired by another operation,

permit the operation to execute when the one or more first locks and the second lock do not conflict with the read-write lock, and

serially execute the operation and the other operation when at least one of the one

or more first locks or the second lock conflicts with the read-write lock acquired by the

other operation.